O-P^3: Supporting the Planning Process using Open Planning Process Panels

John Levine, Austin Tate, and Jeff Dalton, University of Edinburgh

THE AUTHORS INTRODUCE OPEN PLANNING PROCESS PANELS, WHICH ARE BASED ON EXPLICIT MODELS OF THE PLANNING PROCESS AND CAN COORDINATE THE DEVELOPMENT AND EVALUATION OF MULTIPLE COURSES OF ACTION. THIS WORK HAS AN IMPACT ON A NUMBER OF IMPORTANT RESEARCH AREAS OUTSIDE OF PLANNING, INCLUDING WORKFLOW SUPPORT.

REAL-WORLD PLANNING IS A complicated business. It is a multiuser, multiagent collaboration in which teams of people must explore different options to synthesize a solution to given requirements. Specifically, the planning process is the execution of a plan—agents act in parallel, sharing resources, communicating results, and so on. We can make this planning process explicit and use it as a central device for workflow coordination and visualization—we used this idea to create Open Planning Process Panels (O-P^3).

O-P^3 can coordinate the workflow between multiple agents and visualize the development and evaluation of multiple courses of action (COAs). We have used O-P^3 to implement two real applications—the Air Campaign Planning Process Panel (ACP^3) and O-Plan, a two-user, mixed-initiative Web demonstration of planning. In ACP^3, O-P^3 helps build a visualization panel for a complex multiagent planning and evaluation demonstration (TIE 97-1), which uses 11 different software components and involves several users. In O-Plan, O-P^3 technology enables the development and evaluation of multiple COAs by a commander, a planning staff member, and an O-Plan automated planning agent.

O-P^3 technology could impact several important research areas. We envision O-P^3 being used for a planning system in which a team of people and a collection of computer-based planning agents act together to solve a difficult real-world planning problem. Both the human and the system agents act in given roles and are constrained by what they are authorized to do, but they also have the ability to work under their own initiative and volunteer results when this is appropriate. When the planning process is under way, the agents typically work on distinct parts of the plan synthesis in parallel. The agents can also work in parallel to explore different possible courses of action. For example, while one COA is being evaluated, another two might be in the process of being synthesized.

O-P^3 technology

The generic O-P^3 is based on an explicit model of the planning process, which is encoded using an activity modeling language (such as IDEF3) that represents the planning process as a partially ordered network of actions. Some actions have expansions down to a finer level of detail (such as to another partially ordered network).

The purpose of O-P^3 is threefold: to display the planning process's node status to the users, to let users compare the planning process products (such as the COAs), and to control the next steps on the "workflow fringe"—the next possible actions, given the planning process's current status. In the context of creating plans, we designed O-P^3 to allow the development of multiple COAs and the evaluation of those COAs using various plan evaluations.

A generic O-P^3 panel has any of a number of *subpanels*, which we can tailor to support specific users or user roles. These include

- a COA comparison matrix that shows COAs versus elements of evaluation; step status in the planning process; and the outstanding issues for a COA that is being synthesized, which an agent must address before the COA is ready to execute;
- a graphical display showing the planning process's status as a Program Evaluation Review Technique (PERT) chart; and
- other visualizations, such as bar charts, intermediate process product descriptions, and textual plan descriptions.

The generic O-P^3 methodology for building the Open Planning Process Panels consists of these steps, which the software designer carries out:

1. Consider the agents (human and system) involved in the overall planning process, then assign roles and authorities to those agents.
2. Construct a planning-process activity model that shows the partial ordering and decomposition of the actions and which agents can carry out which actions.
3. Build a model of the planning process's current state and an activity monitor, which will update this state model as actions in the planning process take place.
4. Construct appropriate O-P^3 interfaces for each human agent in the planning process, taking into account the role it plays in the interaction. This means that each user role will have an O-P^3 interface that is tailored to the task's overall nature.

Generic O-P^3 design rules inform O-P^3 interface construction. Each user role in the planning process is provided with a panel that the software designer has tailored to that role's activities and needs. The designer then assigns each user role a color to distinguish between the roles. This color serves, for example, as a background for the panel's header. Because a given user might act in more than one distinct user role, this provides a useful visual cue as to which user role is being enacted at any one time.

The generic O-P^3 panel consists of three parts: a graph subpanel (PERT chart), a matrix subpanel (COA comparison matrix), and other subpanels (such as information on assumed environmental conditions). The graph subpanel and the other subpanels are optional items, depending on how useful they are for a given application. The graph subpanel contains a partially ordered graph that shows the planning process's activity model. Because the activity model might be large and might apply for each COA being developed, showing the whole network might not be possible, so the users might need some sort of navigation based on decompositions and switching between COAs.

The actions shown in the graph subpanel are annotated with colors to show their current status in the state model. We have adapted the colors from other ARPI (D*A*RPA/*R*ome *P*lanning *I*nitiative) plan visualization work.[1] The matrix subpanel is a table that contains two types of rows and two types of columns. The rows are process steps (verb phrases) and COA descriptors (noun phrases). The process step labels are colored with the user role background color, and the COA descriptors are white. The columns are the individual COAs being developed (labeled COA-N) and a column reflecting the overall workflow (labeled Overall).

The matrix subpanel's process steps are an appropriately flattened form of the planning process's activity model. O-P^3 can show the action status, using the same colors as in the graph subpanel. Clicking on a hyperlink shows the currently active workflow fringe (what step we can do next). The rows have three sections, running from top to bottom. The first section deals with process steps before plan synthesis, such as setting the COA requirements. The middle section consists of the COA descriptors and is filled out when a COA has been synthesized. The final section consists of process steps that come after plan synthesis, such as addressing any outstanding issues and viewing the resulting COA in various ways.

The COA descriptors relate to the COA products the planning process's steps produce, such as the plan's minimum duration and its effectiveness. Separate plan evaluators, simulators, and so forth can provide the descriptors. Users can select the COA descriptors to show only the critical elements of evaluation. Colors can show whether the result is acceptable and raises no issues (green), is possibly acceptable but has some issues to note (orange), or is not acceptable unless the user relaxes the initial requirements (red).

The other subpanels can contain additional useful information such as tables showing the COA objectives and assumed environmental conditions for each COA.

As mentioned at the start of this article, the O-P^3 agent interfaces let human agents play their parts in the overall planning process, alongside the system agents, which include AI planners, schedulers, plan evaluators, and so on (see Figure 1).

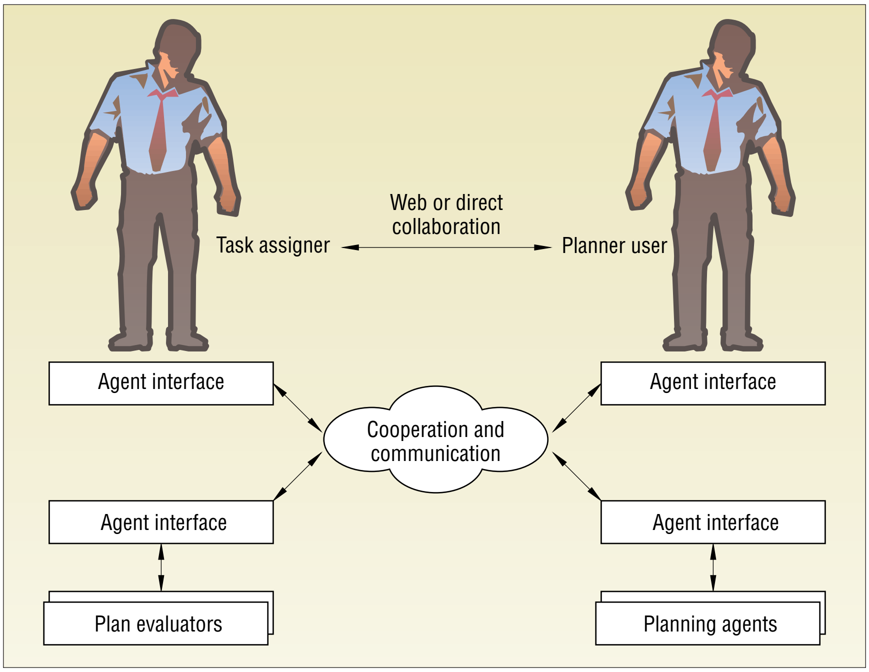


Figure 1. Using O-P^3 interfaces.

Application 1: ACP3

The ARPI TIE 97-1 demonstration brings together 11 separately developed software systems for planning and plan evaluation. When the demonstration runs, these systems work together to create and evaluate multiple courses of action in the domain of air campaign planning. The systems communicate with each other by exchanging KQML messages. In theory, we could discover what's happening at any given time by watching these messages, but this is obviously less than ideal because these messages use technological terms that are far removed from the user community's terminology.

Our aim was to use O-P3 technology to build a visualization component for this demonstration, which would let the target end users view the planning process's current state in terms with which they are familiar. This has resulted in ACP3—the Air Campaign Planning Process Panel.

Modeling the planning process. We can describe TIE 97-1's software components as performing activities such as planning, scheduling, simulation, and plan evaluation. We could discuss hierarchical task network planning and Monte Carlo simulation methods, but end users are more likely to conceive of the processes of air campaign planning in more general, domain-related terms, such as "develop JFACC guidance" and "create support plan." Building models of the planning process, which are traditionally rooted in established ACP terminology, can bridge the gaps in terminology and in description levels. We therefore used the previously elicited and verified ACP process models[2] as our source of terminology and as the basis of our IDEF3 models for TIE 97-1's planning process. The full models we used for building ACP3 appear elsewhere.[3]

Building ACP3. Figure 2 shows the ACP3 viewer. As we stated earlier, ACP3 tracks the overall planning process and displays this to the viewers of the ARPI TIE 97-1 demonstration in a meaningful way using appropriate military process terminology. The planning process appears in two separate subpanels. The tabular COA comparison matrix shows COAs being developed (columns) against a tree-based view of the planning process. The graph viewer subpanel shows the planning process as a PERT network. Because the planning process consists of many nodes with expansions, the graph viewer can only display one graph from the planning process for one COA. Users can reach other graphs by clicking on nodes with expansions, and they can choose which COA to view.

The two views are required because the planning process in TIE 97-1 is complex. You can see the whole process for every COA in the COA matrix, but information about the partial ordering of the actions in a graph is lost when ACP3 converts the graph to a tree structure. The graph viewer shows the full partial ordering, but space considerations mean that the system can show only a single graph for a single COA at one time.

The ACP3 process monitor works by watching for certain KQML messages, which it can relate to the status of certain nodes in the ACP process models. As the demonstration proceeds, the status of actions in the model progress from not yet ready to execute (white), to ready to execute (orange), to executing (green), and finally to complete (blue). The final column in the COA matrix is labeled "Overall" and summarizes the overall status of the COA creation and evaluation process.

The panel is written entirely in Java to form the basis for future Web-based process editors and control panels.

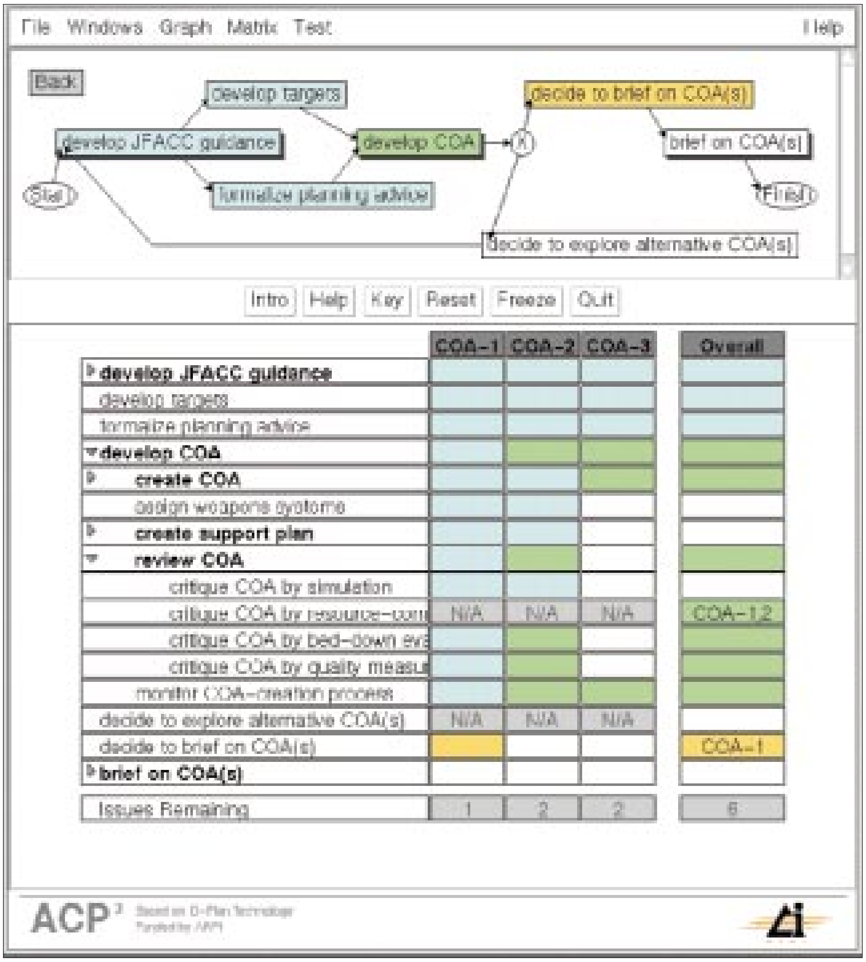


Figure 2 . The ACP3 viewer.

Application 2: O-Plan

The O-Plan project[4,5] is concerned with supporting multiagent mixed-initiative planning. The current O-Plan Web demonstration shows interaction between two human agents and one software planning agent (the O-Plan plan server). The overall concept for our demonstrations of O-Plan as it acts in a mixed-initiative multiagent environment is to have humans and systems work together to populate the

O-P³ interface's COA matrix component.

As Figure 3a shows, we envision two human agents acting in the user roles of *task assigner* and *planner user*, working together to explore possible solutions to a problem and using automated planning aids to do so. Figure 3b shows how the two human agents work together to populate the matrix. The TA sets the requirements for a particular COA (such as what top-level tasks users will perform), selects appropriate evaluation criteria for the resulting plans, and decides which COAs to prepare for briefing. The PU works with O-Plan to explore and refine the different possible COAs for a given set of top-level requirements. The two users can work in parallel, as the example scenario demonstrates.

The overall planning task is thus shared among three agents who act in distinct user and system roles. The TA is a commander who is given a crisis to deal with and who needs to explore some options. This person will receive field reports on the developing crisis and environmental conditions. The PU is a staff member who provides the TA with plans that meet the specified criteria. In doing so, the PU uses the O-Plan automated planning agent, which generates plans for the PU to see. The PU will typically generate a number of possible COAs using O-Plan and will only return the best ones to the TA.

For our current demonstration, we use a general-purpose logistics and crisis operations domain, which is an extension of our earlier Non-Combative Evacuation Operations and logistics-related domains.[6] This domain, together with the O-Plan Task Formalism implementation, is described in detail elsewhere.[5]

Each human user has an O-P³ panel, which is implemented using a CGI-initiated HTTP server in Common Lisp and which can therefore run on any Web browser. The Common Lisp process returns standard HTML pages. This way of working has many advantages:

- each user can use a different type of machine (Unix, PC, Mac) and run a different type of Web browser (Netscape, Internet Explorer, Hotjava, and so on);
- the only requirement for running O-Plan is a Web connection and a Web browser (no additional software installation is needed); and
- the two users can be geographically separate—in this case, voice communication through telephone or teleconferencing is all that is required besides the linked O-P³ interfaces.

Software designers make the planning process for the TA and the PU explicit through the hypertext options displayed in the process parts of the O-P³ panels. The options are *not present* (not ready to run yet), *active* (on the workflow fringe), or *inactive* (completed). Further parts of the planning process are driven by issues that O-Plan or the plan evaluation agents can raise about a plan under construction, and which either or both of the human agents can handle. Because the planning process is made explicit to the two users through these two mechanisms, other visualizations of the planning process are not required. However, the planning process's products (the COAs) are complex artifacts for which multiple views are needed. In the current version, the user can view the COAs as a PERT network, a textual narrative, or a plan-level expansion tree (all at various levels of detail).

The user roles are arranged such that the TA has authority over the PU, who in turn has authority over O-Plan. This means that the TA defines the limits of the PU's activity, and the PU then acts within those bounds to define what O-Plan can do. Other aspects of what the two users are authorized to do are made explicit by the facilities included in their respective panels.

The COA comparison matrix. Figure 4 shows the two panels for the TA and PU. Each user controls the plan evaluation elements (which are shown) to enable the user to choose critical elements of evaluation. In the example scenario given later, the TA is only interested in the minimum duration and the effectiveness, so the TA only selects these. However, the PU wants a variety of data to pick the best COA, so the PU shows all evaluations.

The TA sets up the top-level requirements for a COA. Once this is done, the TA passes the COA across to the PU, whose matrix is initially blank. The PU then explores a range

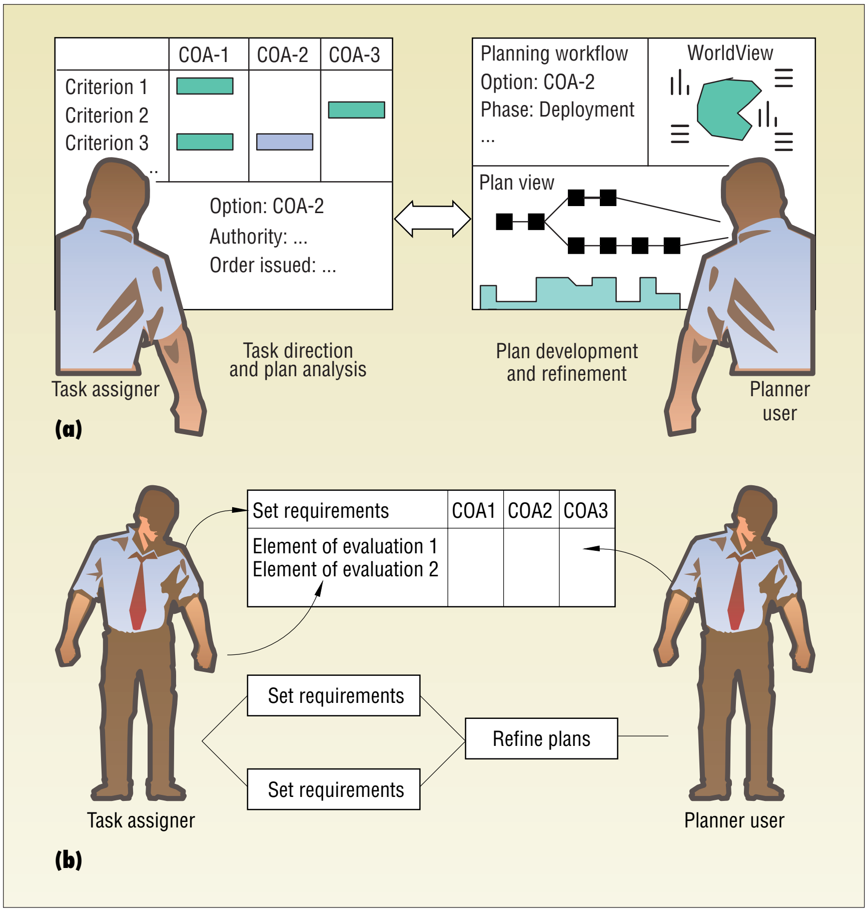


Figure 3. (a) Communication between and (b) the roles of the task assigner and the planner user.

of possible COAs for the specified requirements and returns the best ones to the TA. When the PU returns a COA to the TA, the column for that COA appears in the TA's matrix. The PU and the TA can work in parallel, as we show in the next section.

Demonstration scenario. The O-Plan Web demonstration illustrates mixed-initiative interaction between two human agents and one system-planning agent engaged in developing multiple qualitatively different COAs. O-P^3 interfaces are provided for the two human users that are tailored to their roles. The following scenario illustrates how we envision the system being used.

Initial situation and preparations. The action takes place on the fictional island of Pacifica, with emergencies planned for the cities of Abyss, Barnacle, and Calypso. The TA is told to deal with injured civilians at the three cities within the next 18 hours. Plans are only acceptable if their effectiveness is 75% or greater. The weather forecast gives a 50% chance of a storm within the next 24 hours (see Figure 5a).

The TA sets up the default situation, setting the time limit to 18 hours. The weather and road situations keep their default values, pending more accurate reports.

COA-1. The TA first explores the option of evacuating the injured from all three cities in clear weather. The TA passes the COA requirements directly to the PU. The PU generates a plan is that executes in 12 hours and is 77% effective, which is acceptable. The plan has three issues outstanding. The PU addresses these and returns the plan to the TA.

COA-2. The TA then sets up a second COA with the same evacuation tasks but this time assumes stormy weather to check for all eventualities. The TA passes this new set of COA requirements to the PU. The first plan generated takes 21 hours and is 61% effective, both of which are unacceptable. The PU asks the O-Plan planner for an alternative plan. The new plan (COA-2.2) executes in 16 hours and is 75% effective, both of which are acceptable. The PU returns COA-2.2 to the TA and deletes COA-2.1. At this point, the TA has an acceptable plan for both clear and stormy conditions.

Developing situation. The Barnacle field station now contacts the TA. Reports are coming in of an explosion at the power station, causing a gas leak. This is believed to be due to a terrorist bomb, so it seems wise to fix the gas leak and send in a bomb squad to defuse any remaining bombs. Meanwhile, the latest weather report indicates that a storm is brewing and has a 95% chance of hitting the island (see Figure 5b).

COA-2.2.2. To deal with this turn of events, the TA splits COA-2.2 (the realistic weather assumption) into two suboptions and adds two new tasks to COA-2.2.2—to repair the gas leak at Barnacle and send a bomb squad to Barnacle. COA-2.2.2 is now passed to the PU. Because the original COA-2.2 took 16 hours, the PU turns the schema choice on, to have fine control of the two new tasks added to the existing plan. The PU has the option of using fast or slow vehicles for the two tasks and chooses fast vehicles. However, this plan takes 22 hours and is 63% effective. The PU replans and chooses a mixture of fast and slow vehicles for the "repair gas leak" task and a fast vehicle for the "defuse terrorist bomb" task. Although better,

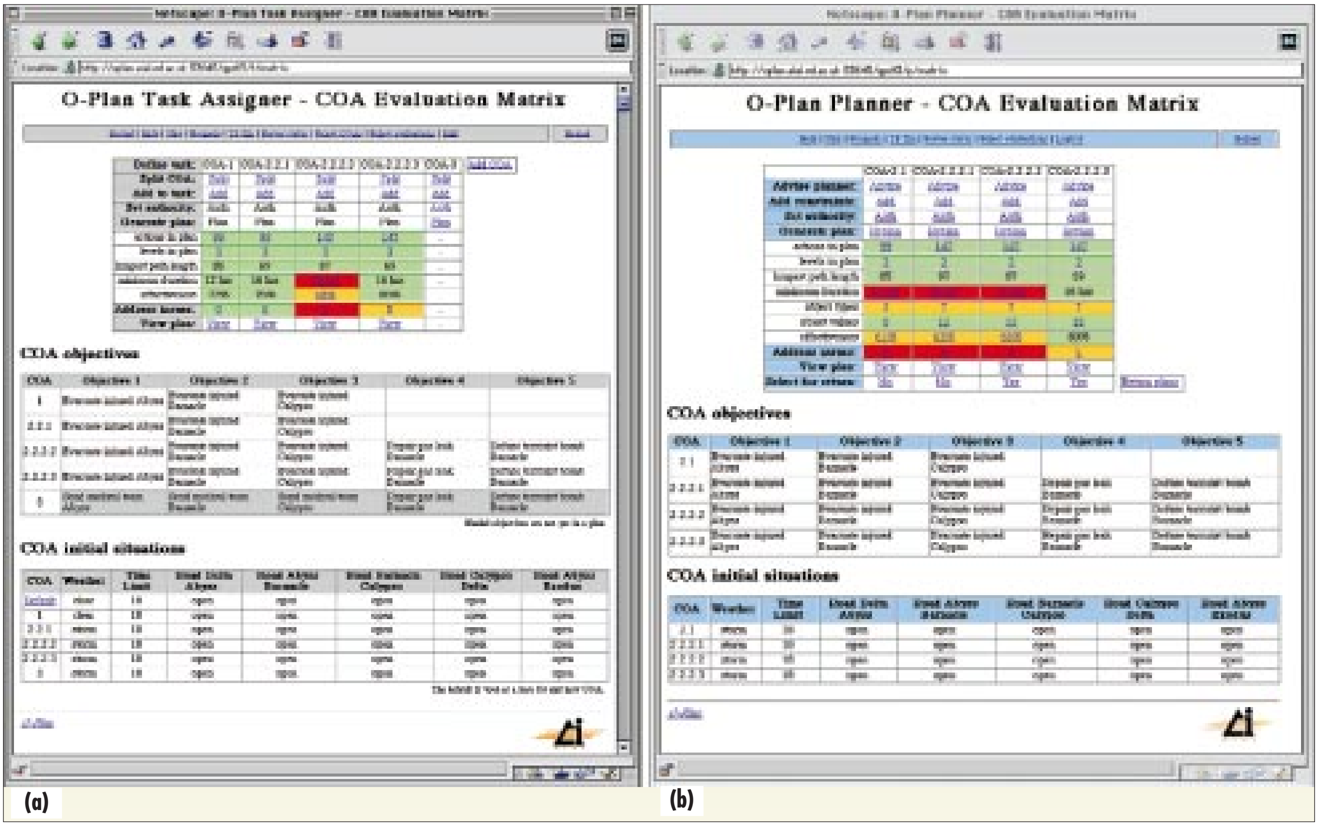


Figure 4. The (a) task assigner's panel and (b) planner user's panel.

the new plan now takes 19 hours and is 68% effective. The TA is getting impatient and tells the PU, "This is taking too long. Just give me the best one so far." The PU returns COA-2.2.2.2, keeping COA-2.2.2.1 for further back-office work.

COA-3. The TA decides to send medical teams to the three cities to deal with the injured civilians rather than evacuating them. After updating the default situation to reflect the weather report, the TA starts to set up COA-3 with these tasks, and so begins to define the requirements on the screen.

COA-2.2.2.3. Meanwhile, the PU has continued to explore the possibilities for COA-2.2.2. The plan improved when the PU used some slow vehicles in the plan, most likely because the limited number of fast vehicles are repeatedly in use, resulting in a longer (more linear) plan. The PU presses "Replan" and chooses to use a slow vehicle in the "defuse terrorist bomb" task. Sending the bomb squad is only a precaution—using the limited number of fast vehicles for evacuating the injured and fixing the known gas leak seems like a good idea. The PU is right—the resulting plan executes in 16 hours and is 80% effective. Viewing the plan shows that this plan has good parallelism. The PU now addresses the issues raised by COA-2.2.2.3 and returns this plan to the TA, saying, "I think I've fixed the problem with COA-2.2.2."

Back to COA-3. The TA sees the new plan: "This looks good; now see what you can do with COA-3 as an alternative." The PU (still in the "ask user" schema selection mode) selects the fast vehicle option for four of the tasks but selects a slow vehicle for the "defuse terrorist bomb" task. The resulting plan executes in 12 hours and is 79% effective.

Choice of COA. The TA now has a choice between COA-2.2.2.3 and COA-3. Although COA-3 takes four hours less, it is slightly less effective; more important, it only sends medical teams to the three cities rather than evacuating the injured people. The TA could now examine other details of the two plans, using the plan views and the other elements of evaluation, to make an informed choice between the two or plan further.

THE ACP^3 AND THE O-PLAN WEB demonstration of crisis response planning have an explicit planning-process notion: multiagent interaction. The agents in both systems have roles that relate to the actions users can carry out in the planning process. Both systems use a COA matrix, which shows possible steps in the planning process for each course of action being developed. In ACP^3, we use this as a visualization device. In the O-Plan demonstration, the population of this matrix is central to the mixed-initiative interaction between the TA, PU, and O-Plan.

An O-P^3 process panel is being built as part of the Coalition Agents Experiment (CoAX) under DARPA's Control of Agent Based Systems program. This panel will include the matrix subpanel and the graph subpanel from ACP^3. However, the CoAX panel will likely include new subpanels to provide a "process product" perspective (showing the status of various information products under development) and new subpanels that give more role-specific workflow status for a number of user types. The main innovation in the CoAX panel will feature hooks to allow AI planning technology for dynamically generating and adapting the planning process to accommodate changing requirements and situations. We have already demonstrated such an intelligent workflow planning aid using O-Plan for the air campaign planning process.[7]

Acknowledgments

DARPA and the US Air Force Research Laboratory at Rome (AFRL) supported this work under grant number F30602-95-1-0022. The US Government and the University of Edinburgh are authorized to reproduce and distribute reprints for their purposes, notwithstanding any copyright annotation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing official policies or endorsements, either express or implied, of DARPA, AFRL, the US Government, or the University of Edinburgh.

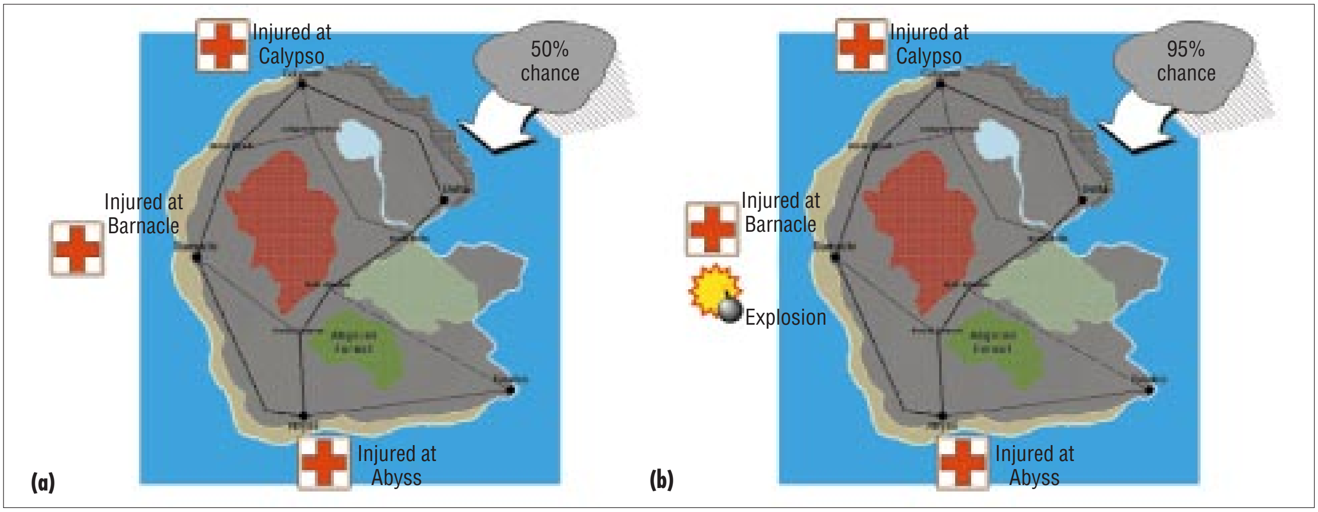
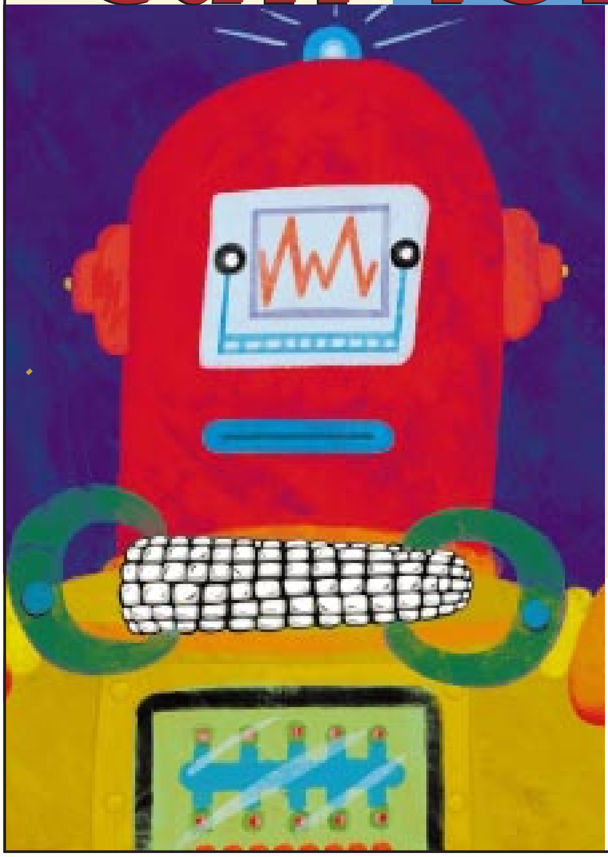


Figure 5. The (a) initial and (b) developing situations.



References

1. J. Stillman and P. Bonissone, “Technology Development in the ARPA/RL Planning Initiative,” *Advanced Planning Technology*, A. Tate, ed., AAAI Press, Menlo Park, Calif., 1996, pp. 10–23.

2. B. Drabble, T. Lydiard, and A. Tate, *Process Steps, Process Product and System Capabilities*, Initiative Support and ACPT Testbed Tech. Report ISAT-AIAI/TR/4, AI Application Inst., Univ. of Edinburgh, 1997.

3. S. Aitken and A. Tate, *Process Modelling of the TIE 97-1 Demonstration: Modelling Complex Techniques Using ACP Terminology*, ISAT Tech. Report ISAT-AIAI/TR/6, AI Application Inst., Univ. of Edinburgh, 1997.

4. B. Drabble, A. Tate, and J. Dalton, *ACP Process Management: O-Plan IFD-5 Qualifier*, O-Plan Tech. Report ARPA-RL/O-Plan/TR/30, AI Application Inst., Univ. of Edinburgh, 1996.

5. A. Tate, J. Dalton, and J. Levine, “Generation of Multiple Qualitatively Different Plan Options,” *Proc. Fourth Int’l Conf. AI Planning Systems (AIPS ’98)*, AAAI Press, Menlo Park, Calif., 1998, pp. 27–34.

6. G.A. Reece et al., “The PRECIS Environment,” paper presented at the ARPA-RL Planning Initiative Workshop at AAAI ’93, Washington D.C., July 1993; www.aiai.ed.ac.uk/~oplan/documents/1993/93-arpi-precis.ps (current Oct. 2000).

7. A. Tate, B. Drabble, and J. Dalton, “O-Plan: A Knowledge-Based Planner and Its Application to Logistics,” *Advanced Planning Technology*, A. Tate, ed., AAAI Press, Menlo Park, Calif., 1996, pp. 259–266.

John Levine is a senior research fellow at the University of Edinburgh’s Artificial Intelligence Applications Institute, working on AI planning systems and natural language generation. His current research program consists of interfaces to planning systems, issue-based and mixed-initiative planning systems, and the application of genetic algorithms and genetic programming techniques to planning problems. He has an MA in computer science, an MPhil in computer speech and language processing, and a PhD in natural language processing, all from the University of Cambridge. Contact him at AIAI, Div. of Informatics, Univ. of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; j.levine@ed.ac.uk.

Austin Tate is technical director of AIAI and holds the Personal Chair of Knowledge-Based Systems at the University of Edinburgh. As well as engaging in the research, development, and application of knowledge-based methods, he also has a background in databases and software engineering. He graduated from the University of Lancaster with a degree in computer science and has a PhD in machine intelligence from the University of Edinburgh. He is a Fellow of the Royal Society of Edinburgh, a Chartered Engineer, and an elected Fellow of the AAAI. Contact him at AIAI, Div. of Informatics, Univ. of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; a.tate@ed.ac.uk.

Jeff Dalton is a senior computer scientist at AIAI. His research interests include AI planning algorithms, programming languages, and Web-based systems. He received an AB in mathematics from Dartmouth College and was a member of X3J13, the technical committee that developed the ANSI Common Lisp standard. Contact him at AIAI, Div. of Informatics, Univ. of Edinburgh, 80 South Bridge, Edinburgh EH1 1HN, UK; j.dalton@ed.ac.uk.